IN THE CLAIMS:

Please cancel claim 39, and amend the claims as follows:

1. (Currently Amended) A method for tracking at least one process using a socket object, where the at least one process is utilized to execute an application program, the method comprising:

creating a process list for the socket object, where the process list contains a process identifier for a first process using the socket object; ~~and~~

updating, if a second process is using the socket object, the process list to include the process identifier for the second process; and

determining if the process list is empty, and if the process list is empty, deleting the socket object.

2. (Original) The method of claim 1 wherein the process list is displayed on a user interface in response to a user interface command entered by a user.

3. (Original) The method of claim 1 wherein the step of updating comprises:

adding the process identifier of the second process to the process list if the second process is to use the socket object.

4. (Original) The method of claim 3 wherein the second process is to use the socket object if a socket descriptor created for the socket object is passed from the first process to the second process.

5. (Original) The method of claim 1 wherein the step of updating comprises:

removing the process identifier of at least one of the first process and second process from the process list if the at least one of the first process and second process no longer uses the socket object.

6. (Original) The method of claim 5 wherein the at least one of the first process and second process no longer uses the socket object if a socket descriptor created for the socket object is removed from the at least one of the first process and second process.

7. (Original) The method of claim 1 wherein the step of updating comprises:

removing the process identifier of at least one of the first process and the second process from the process list if the at least one of the first process and second process expires.

8. (Original) The method of claim 1 wherein the first process comprises a Sockets Application Program Interface (API) function utilized to create the socket object.

9. (Original) The method of claim 8 wherein the Sockets API function comprises one of a socket ( ) function, a socketpair ( ) function and an accept ( ) function.

10. (Original) The method of claim 1 wherein the creating and updating are performed by an operating system after a computer executes a sockets support program.

11. (Original) The method of claim 1 wherein the first process and the second process are provided in the same computer system.

12. (Original) The method of claim 1 wherein the first process and the second process are provided in different computer systems.

13. (Original) The method of claim 1 wherein the process identifier comprises at least one of a process name, a user name associated with the process name and a process number.

14-17. (Cancelled)

18. (Currently Amended) An apparatus for tracking at least one process of an application program using a socket object, the apparatus comprising:

a memory for storing an operating system and a sockets support program; and

a processor, coupled to the memory, for performing a method upon executing the sockets support program retrieved from the memory, the method comprising:

creating the process list for a socket object, where the process list contains a process identifier for a first process using the socket object, ~~and~~

updating, if a second process is using the socket object, the process list to include the process identifier of the second process, and

determining, by the socket object, if the process list is empty, and if the process list is empty, deleting the socket object.

19. (Original) The apparatus of claim 18 further comprising:

a network interface for coupling the socket object with a remote device.

20. (Original) The apparatus of claim 18 further comprising:

a display device, coupled to the processor, for displaying the process list when the processor retrieves and executes a user interface program from the memory.

21. (Original) The apparatus of claim 18 wherein the operating system comprises one of UNIX, IBM AIX, IBM OS/400 and Microsoft Windows.

22. (Original) The apparatus of claim 18 wherein the network interface couples the first process to the second process.

23. (Currently Amended) A computer readable medium storing a software program that, when executed by a processor of a computer, causes the computer to perform a method comprising:

creating a process list for a socket object, where the process list contains a process identifier for a first process using the socket object; ~~and~~

updating, if the second process is using the socket object, the process list to include the process identifier of the second process, and

determining, by the socket object, if the process list is empty, and if the process list is empty, deleting the socket object.

24. (Original) The computer readable medium of claim 23 wherein the process list is displayed on a user interface in response to a user interface command entered by a user.

25. (Original) The computer readable medium of claim 23 wherein the step of updating comprises:

adding the process identifier of the second process to the process list if the second process is to use the socket object.

26. (Original) The computer readable medium of claim 25 wherein the second process is to use the socket object if a socket descriptor created for the socket object is passed from the first process to the second process.

27. (Original) The computer readable medium of claim 23 wherein the step of updating comprises:

removing the process identifier of at least one of the first process and second process from the process list if the at least one of the first process and second process no longer uses the socket object.

28. (Original) The computer readable medium of claim 27 wherein the at least one of the first process and second process no longer uses the socket object if a socket descriptor created for the socket object is removed from the at least one of the first process and second process.

29. (Original) The computer readable medium of claim 23 wherein the step of updating comprises:

removing the process identifier of at least one of the first process and the second process from the process list if the at least one of the first process and second process expires.

30. (Original) The computer readable medium of claim 23 wherein the first process comprises a Sockets Application Program Interface (API) function utilized to create the socket object.

31. (Original) The computer readable medium of claim 30 wherein the Sockets API function comprises one of a socket ( ) function, a socketpair ( ) function and an accept ( ) function.

32. (Original) The computer readable medium of claim 23 wherein the creating and updating are performed by an operating system after a computer executes a sockets support program.

33. (Original) The computer readable medium of claim 23 wherein the first process and the second process are provided in the same computer system.

34. (Original) The computer readable medium of claim 23 wherein the first process and the second process are provided in different computer systems.

35. (Original) The computer readable medium of claim 23 wherein the process identifier comprises at least one of a process name, a user name associated with the process name and a process number.

36-38. (Cancelled)

39. (Cancel)

Please add the following new claims:

40. (New) A method for tracking at least one process using a socket object, the method comprising:

creating a process list for the socket object, where the process list contains a process identifier for the first process using the socket object;

providing a socket descriptor for the socket object;

adding a second process identifier for a second process to the process list if the socket descriptor is passed to the second process;

determining if the socket descriptor passed to the second process has been removed from the second process, and if so removing the second process identifier for the second process from the process list;

determining if the second process has expired, and if so removing the second process identifier for the second process from the process list; and

determining, by the socket object, if the process list is empty, and if the process list is empty, deleting the socket object.

(41) (New) The method of claim 1 wherein the determination of whether the process list is empty and the deletion of the socket object is performed by the socket object.